

REPORT



과목명		빅데이터분석활용
담당교수		이승천 교수님
학과		응용통계학과
학년		4
학번		201452024
이름		박상희
제출일		2019년 11월 4일

Fibonacci Sequence

수학에서, 피보나치 수(영어: Fibonacci numbers)는 첫째 및 둘째 항이 1이며 그 뒤의 모든 항은 바로 앞 두 항의 합인 수열이다. 처음 여섯 항은 각각 1, 1, 2, 3, 5, 8이다. 편의상 0번째 항을 0으로 두기도 한다.

Python Code

Fibonacci Sequence

```
def Fibonacci(n) :
    if n == 1 or n == 2 :
        return 1
    else :
        return Fibonacci(n-1) + Fibonacci(n-2)
```

Print Sequence

```
for i in range(1,11,1) :
    if i == 1 :
        print('Fibonacci Sequence')
    print("{} ".format(Fibonacci(i)), end=" ")
```

=====

Fibonacci Sequence

1 1 2 3 5 8 13 21 34 55

Python Code With Ignition

$$a_1 = 1, a_2 = 1, a_{n+2} = a_{n+1} + a_n$$

점화식을 구하기 위해 위의 일반항을 변형한다.

$$a_{n+2} - \alpha a_{n+1} = \beta(a_{n+1} - \alpha a_n)$$

$$a_{n+2} - \alpha a_{n+1} = \beta(a_{n+1} - \alpha a_n) = \beta^2(a_n - \alpha a_{n-1}) = \dots = \beta^n(a_2 - \alpha a_1) = \beta^{n+1}$$

$$\therefore a_n - \alpha a_{n-1} = \beta^{n-1} \quad *** \text{ ① }, \quad \alpha \text{ 와 } \beta \text{ 를 바꾸어 생각해드 결과는 같다.}$$

$$\therefore a_n - \beta a_{n-1} = \alpha^{n-1} \quad *** \text{ ②}$$

$$\text{①과 ②를 연립하여 풀면, } a_n = \frac{1}{\beta - \alpha}(\beta^n - \alpha^n) \text{ 이고,}$$

$$\alpha + \beta = 1, \alpha\beta = -1 \text{ 이므로 } \alpha \text{ 와 } \beta \text{ 는 이차방정식 } x^2 - x - 1 = 0 \text{ 의 두 근이다.}$$

$$\therefore \alpha = \frac{1 - \sqrt{5}}{2}, \beta = \frac{1 + \sqrt{5}}{2} \text{ 이고, 일반항은 } a_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right) \text{ 이다.}$$

=====

def Fibonacci_Ignition() :

```
    result = []
```

```
    for n in range(1,21,1) :
```

```
        ROOT_FIVE = 5 ** 0.5
```

```
        SEQUENCE = (1/(ROOT_FIVE)) * (((1+ROOT_FIVE)/2)**n - ((1-ROOT_FIVE)/2)**n)
```

```
        SEQUENCE = int(SEQUENCE)
```

```
        result.append(SEQUENCE)
```

```
    return result
```

```
print(Fibonacci_Ignition())
```

=====

[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]

Summary Method

```
# Data 생성
from scipy.stats import norm
from pip._vendor.urllib3 import add_stderr_logger
x = [norm.rvs() * 5 + 170. for _ in range(100)]

# Summary Function
def SUMMARY(x) :

    # 결과 저장소
    result = []

    # Sort Function
    def sort_function (X) :
        N = len(X)
        for i in range(N-1):
            k = i
            for j in range(i+1,N) :
                if X[k] > X[j] :
                    k = j
            X[k],X[i] = X[i],X[k]
        return X

    # 정렬 실행
    x = sort_function(x)

    # 데이터의 길이
    N = len(x)

    # 데이터들의 합
    def SUM_FUNCTION(x) :
        SUM = 0
        for i in range(N) :
            SUM = SUM + x[i]
        return SUM
    SUM = SUM_FUNCTION(x)

    # 데이터들의 제곱 합
    def SUM_2_FUNCTION(x) :
        SUM_2 = 0
        for i in range(N) :
            SUM_2 = SUM_2 + x[i]*x[i]
        return SUM_2
    SUM_2 = SUM_2_FUNCTION(x)

    # 최솟값
    def MIN_FUNCTION(x) :
        return x[0]
    MIN = MIN_FUNCTION(x)
    result.append(MIN)

    # 최댓값
    def MAX_FUNCTION(x) :
        return x[N-1]
    MAX = MAX_FUNCTION(x)
    result.append(MAX)

    # 평균
    def MEAN_FUNCTION(x) :
        return SUM / N
    MEAN = MEAN_FUNCTION(x)
    result.append(MEAN)

    # 중앙값
    NN = int((N+1)/2)
    def MEDIAN_FUNCTION(x) :
        return x[NN]
    MEDIAN = MEDIAN_FUNCTION(x)
    result.append(MEDIAN)

    # 분산
    def VAR_FUNCTION(x) :
        return SUM_2/N - MEAN*MEAN
    VAR = VAR_FUNCTION(x)
    result.append(VAR)

    # 표준편차
    def STD_FUNCTION(x):
        return VAR**0.5
    STD = STD_FUNCTION(x)
    result.append(STD)

    # 결과 출력
    NAME = [ "MIN", "MAX", "MEDIAN", "MEAN", "VAR", "STD" ]
    print("== Summary of X ==")
    for i in range(6) :
        print("{}: {:.4f}".format(NAME[i], result[i]))

    # 결과 반환
    return result
```

데이터가 주어졌을 때 그 데이터의 최솟값, 최댓값, 중앙값, 평균, 분산, 표준편차를 리스트로 반환하고 출력한다.

프로그램 실행 결과

```
== Summary of X ==
MIN: 159.7091
MAX: 180.1581
MEDIAN: 169.7530
MEAN: 172.7990
VAR: 18.8938
STD: 4.3467
```